Riemannian geometry applied to BCI classification

Alexandre Barachant[1], Stéphane Bonnet[1], Marco Congedo[2], and Christian Jutten[2]

[1] CEA, LETI, DTBS/STD/LE2S,
17 rue des Martyrs, F-38054 Grenoble, France
[2] Team ViBS (Vision and Brain Signal Processing),
GIPSA-lab, CNRS, Grenoble Universities.
Domaine Univiversitaire, 38402 Saint Martin d'Hères, France

Abstract. In brain-computer interfaces based on motor imagery, covariance matrices are widely used through spatial filters computation and other signal processing methods. Covariance matrices lie in the space of Symmetric Positives-Definite (SPD) matrices and therefore, fall within the Riemannian geometry domain. Using a differential geometry framework, we propose different algorithms in order to classify covariance matrices in their native space.

1 Introduction

A Brain-Computer Interface (BCI) aims at providing an alternative non-muscular communication path and a control system for the individuals with heavy motor disabilities like in Spinal Chord Injury (SCI) or Locked-In Syndrome (LIS) patients. This new interface should perform an automatic decoding of measured brain activity [1].
Non-invasive BCIs use mainly ElectroEncephaloGraphic (EEG) activity recorded from a cap of scalp electrodes [2]. The goal is to detect and classify some specific patterns of EEG activities so as to drive an external effector (e.g. computer mouse, wheelchair, ...) [1]. Different paradigms can be used to activate these brain patterns, either synchronously (evoked potentials) [3] or asynchronously (brain rhythm modulation) after a co-adaptive learning phase. EEG-based features are usually related to the power (or variance) of relevant EEG channels in specific frequency bands (e.g. mu brain oscillation in the frequency band 5-15 Hz) [3]. In this article, we propose a new signal processing framework in BCI applications, which is based on **Riemannian geometry**.

As it will be shown, interesting properties may be derived by considering the space of symmetric positive-definite (SPD) matrices. The main motivation of our work is to make use of the concept of Riemannian distance between SPD matrices in BCI applications. We will introduce basic tools to manipulate EEG data in this Riemannian manifold and illustrate these concepts with a simple and didactic binary classification task. Doing so, EEG data can be manipulated in a convenient way through their spatial covariances, and then detection/classification

can be achieved by measuring, for instance, the Riemannian distance between covariance matrices of signal epochs and and covariance matrices of reference epochs. Indeed, classical methods treats the SPD matrices as if they were naturally lying in the Euclidean space, whereas the natural geometry to be considered is the Riemann geometry.

This approach has already been followed in diffusion tensor imaging to study the statistical properties of a population of geometric objects [4], in image processing to detect pedestrians images [5] or in radar detection and High Resolution Doppler Imagery to achieve a robust statistical estimation of Toeplitz Hermitian positive definite covariance matrices of sensor data time series [6]. First we will present the basic tools of Riemannian geometry in space of SPD matrices, next we will propose some classification and filtering algorithms and finally we will show results on real EEG data.

2 Differential Geometry in Space of SPD Matrices

2.1 Definitions and Properties

This section exposes some **definitions** and **properties** of differential geometry in the Riemannian manifold of SPD matrices. More sophisticated explanations can be found in the reference [7].
A Riemannian manifold is a differentiable manifold in which the tangent space at each point is a finite-dimensional Euclidean space. We denote by $S(n) = \{\mathbf{S} \in M(n), \mathbf{S}^T = \mathbf{S}\}$ the space of all $n \times n$ symmetric matrices in the space of square matrices and denote by $P(n) = \{\mathbf{P} \in S(n), \mathbf{P} > 0\}$ the set of all $n \times n$ symmetric positive-definite (SPD) matrices. The Riemannian distance between two SPD matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ in $P(n)$ is given by [7] :

$$\delta_R(\mathbf{P}_1, \mathbf{P}_2) = \|\mathrm{Log}\left(\mathbf{P}_1^{-1}\mathbf{P}_2\right)\|_F = \left[\sum_{i=1}^{n} \log^2 \lambda_i\right]^{1/2} \tag{1}$$

where the λ_i's are the real strictly positive eigenvalues of $\mathbf{P}_1^{-1}\mathbf{P}_2$ and $\|.\|_F$ is the Frobenius norm of a matrix. In Eq (1),the operator Log (.) is the logarithm of a matrix. Given that SPD matrices are diagonalizable and invertible, Log $(\mathbf{P})$ can be computed by diagonalization of $\mathbf{P}$: Log $(\mathbf{P}) = \mathbf{V}\log(\mathbf{D})\mathbf{V}^{-1}$ with $\log(\mathbf{D})$ the logarithm of each element of $\mathbf{D}$ where $\mathbf{D}$ and $\mathbf{V}$ are respectively the diagonal matrix of eigenvalues and the matrix of eigenvectors of $\mathbf{P}$. The geometric mean in the Riemannian sense, i.e. associated with the metric defined in Eq (1), of m given SPD matrices $\mathbf{P}_1, \ldots, \mathbf{P}_m$ is defined as [7] :

$$\mathfrak{G}\left(\mathbf{P}_1, \ldots, \mathbf{P}_m\right) = \underset{\mathbf{P} \in P(n)}{\operatorname{argmin}} \sum_{i=1}^{m} \delta_R^2\left(\mathbf{P}, \mathbf{P}_i\right) \tag{2}$$

There is no closed-form expression for such mean computation, but iterative algorithms can be employed, as demonstrated in section 2.2.

The shortest path between two points in the Riemannian space of SPD matrices is defined by the geodesic $\gamma(t)$ with $t \in [0,1]$:

$$\gamma(t) = \mathbf{P}_1^{1/2} \left(\mathbf{P}_1^{-1/2} \mathbf{P}_2 \mathbf{P}_1^{-1/2} \right)^t \mathbf{P}_1^{1/2} \tag{3}$$

Identically to the matrix logarithm, the power of SPD matrices can be computed using a diagonalization : $\mathbf{P}^t = \mathbf{V}\mathbf{D}^t\mathbf{V}^{-1}$.

2.2 Tangent Space

In complete Riemannian space, given a point $\mathbf{P} \in P(n)$, it is possible for every point $\mathbf{P}_i \in P(n)$, to identify a tangent vector $\mathbf{S}_i \in S(n)$ such as $\mathbf{S}_i = \dot{\gamma}(0)$ with $\gamma(t)$ the geodesic between $\mathbf{P}$ and $\mathbf{P}_i$. The Riemannian Log map operator $\text{Log}_{\mathbf{P}} : P(n) \rightarrow S(n)$ achieves the mapping $\text{Log}_{\mathbf{P}}(\mathbf{P}_i) = \mathbf{S}_i$.

$T_{\mathbf{P}}$, the tangent space at $\mathbf{P}$, is the space defined by the whole set of tangent vector $\mathbf{S}_i$, here $S(n)$. In this tangent space, the metric is flat and allows us to use arithmetic mean and other classical tools. The Riemannian Exp map operator $\text{Exp}_{\mathbf{P}}(\mathbf{S}_i) = \mathbf{P}_i$ allows to go back in the original space of SPD matrices $P(n)$ in a one-to-one mapping. Both operators are crucial in the manipulation of SPD matrices as we will discover. Using the affine-invariant metric [7], we have the expressions:

$$\text{Exp}_{\mathbf{P}}(\mathbf{S}_i) = \mathbf{P}^{1/2} \text{Exp} \left(\mathbf{P}^{-1/2} \mathbf{S}_i \mathbf{P}^{-1/2} \right) \mathbf{P}^{1/2} \tag{4}$$

$$\text{Log}_{\mathbf{P}}(\mathbf{P}_i) = \mathbf{P}^{1/2} \text{Log} \left(\mathbf{P}^{-1/2} \mathbf{P}_i \mathbf{P}^{-1/2} \right) \mathbf{P}^{1/2} \tag{5}$$

We can refer to [4] for efficient computation. Figure 1 illustrates these operations.

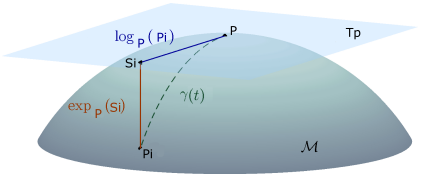

Fig. 1. Tangent space of the manifold $\mathcal{M}$ at point $\mathbf{P}$, $\mathbf{S}_i$ the tangent vector of $\mathbf{P}_i$ and $\gamma(t)$ the geodesic between $\mathbf{P}$ and $\mathbf{P}_i$.

The mean of m SPD matrices can be obtained using the concept of tangent space. Using Riemannian Log map, we first project the whole dataset in tangent space. In this Euclidean space, the arithmetic mean is the correct average estimate and can be easily computed. Finally we project the obtained arithmetic mean into SPD space using Riemannian exponential map. After few iterations, we obtain the geometric mean of SPD matrices. Algorithm 1 explains this process.

Algorithm 1 Mean of m SPD matrices

Input: Ω a set of m SPD matrices $\mathbf{P}_i \in P(n)$ and $\epsilon > 0$.
Output: $\mathbf{P}_\Omega$ the estimated mean in $P(n)$.

1: Initialise $\mathbf{P}_\Omega^{(1)} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{P}_i$
2: **repeat**
3: $\quad \mathbf{S} = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Log}_{\mathbf{P}_\Omega^{(t)}}(\mathbf{P}_i)$ {Arithmetic mean in tangent space}
4: $\quad \mathbf{P}_\Omega^{(t+1)} = \mathrm{Exp}_{\mathbf{P}_\Omega^{(t)}}(\mathbf{S})$
5: **until** $\|\mathbf{S}\|_F < \epsilon$

3 Classifying on Riemannian Manifold

We denote by $\mathbf{E} \in R^{n \times t}$ a given EEG recording epoch with n electrodes and t samples. The spatial sample covariance matrix is proportional to : $\mathbf{P} = \mathbf{E}\mathbf{E}^T$ for centered-data matrix is, by construction, lying in $P(n)$. The goal in BCI is to determine what mental task is associated with a segment of data, or in other terms, determine the class $\omega_i \in \{1, 2\}$ of the observation $\mathbf{E}_i$ or equivalently its spatial covariance matrix $\mathbf{P}_i$. In motor imagery BCI paradigm, the mental task would be for instance, the recognition of either left- and right-hand imagery movements [8]. This binary classification problem is usually tackled using supervised learning where the patient undergoes first a training session where the correspondence $\{\mathbf{E}_i, \omega_i\}$ is known by experiment design, so as to produce decision rules for next unknown test sessions [2]. Taking into account covariance matrices as elements of Riemannian space of SPD matrices, spatial information is directly accessible and classification can be performed without preprocessing, in our approach. Based on the concept presented in section 2, a very simple algorithm, given in Algorithm 2, is proposed for illustration purpose. It is merely based on the computation of Riemannian distances to classify a new epoch.

Algorithm 2 Simple classification based on Riemannian distance

Input: Ω a set of m SPD matrices $\mathbf{P}_i \in P(n)$.
Input: $\omega_i \in \{1, 2\}$ the class of $\mathbf{P}_i$.
Input: $\mathbf{P}_x$ a SPD matrix of unknown class.
Output: ω_x the estimated class of test covariance matrix $\mathbf{P}_x$.

1: $\mathbf{P}_{\Omega_1} = \mathfrak{G}(\mathbf{P}_i)$ with $\{i|\omega_i = 1\}$ {Riemannian mean for class 1}
2: $\mathbf{P}_{\Omega_2} = \mathfrak{G}(\mathbf{P}_i)$ with $\{i|\omega_i = 2\}$ {Reimannian mean for class 2}
3: $d = \delta(\mathbf{P}_x, \mathbf{P}_{\Omega_1}) - \delta(\mathbf{P}_x, \mathbf{P}_{\Omega_2})$
4: **if** $d \leq 0$ **then**
5: $\quad \omega_x = 1$
6: **else**
7: $\quad \omega_x = 2$
8: **end if**
9: **return** ω_x

As it can be observed, the whole algorithm 2 relies on the computation of both intra-class SPD means ($\mathbf{P}_{\Omega_1}, \mathbf{P}_{\Omega_2}$) and the shortest Riemannian distance between the test covariance matrix and the two intra-class SPD means. The main limitation of this approach is that it may exist a large part of distance between the two matrices which is not class-related i.e. the class-related information contained in distance can vanish in front of noise.

Therefore, it is preferable to perform some filtering over SPD matrices before applying Algorithm 2. Our approach is inspired by the principal geodesics analysis (PGA) method of Fletcher et. al. [4]. We search first the geodesics that support class-related information and perform filtering along this line in Riemann space to discard irrelevant information. To compute these filters, we propose a supervised algorithm named FGDA for Fisher Geodesic Discriminant Analysis. This algorithm is an extension of Fisher Linear Discriminant Analysis to the tangent space and is given by Algorithm 3.

Algorithm 3 FGDA Filters

Inputs: Ω a set of m SPD matrices $\mathbf{P}_i \in P(n)$, $\omega_i \in \{1, 2\}$ the class of $\mathbf{P}_i$, K number of selected components.
Outputs: $\widetilde{\mathbf{W}}_k, k = 1, \dots K \in \Re^{n(n+1)/2}$, $\mathbf{P_\Omega}$.

1: $\mathbf{P}_\Omega = \mathfrak{G}(\mathbf{P}_i)$ {Compute Riemannian mean of the whole set}
2: **for** $i = 1$ to m **do**
3: $\quad \mathbf{S}_i = \mathrm{Log}_{\mathbf{P}_\Omega}(\mathbf{P}_i)$ {Apply Riemannian Log map}
4: $\quad \widetilde{\mathbf{S}}_i = vec(\mathbf{S}_i)$ {Keep upper triangular matrix in vector form $n(n+1)/2$}
5: **end for**
6: $\widetilde{\mathbf{W}} = LDA(\widetilde{\mathbf{S}}_i)$ {compute the projection vectors using the Fisher LDA criterion}
7: Select the first K vectors $\widetilde{\mathbf{W}}_k$

In order to project data in tangent space, we compute the Riemannian mean $\mathbf{P}_\Omega$. $\mathbf{P}_\Omega$ is the point where the tangent vectors $\mathbf{S}_i$ will be computed, different points can be used [5], however the use of the Riemannian mean minimizes the approximation caused by the projection in the tangent space of the dataset. After performing data projection into the tangent space, the step 6 of this algorithm compute the different projection vectors using the Fisher LDA criterion [9]. This is a maximisation of the ratio of the *between-class* scatter matrix $\mathbf{\Sigma}_b$ and the *within-class* scatter matrix $\mathbf{\Sigma}_w$ and can be solved easily by eigenvector decomposition of $\mathbf{\Sigma}_w^{-1}\mathbf{\Sigma}_b$ [9]. Interestingly, the exponential map of these filters gives the main geodesics issued from $\mathbf{P}_\Omega$.

Typically, the number K of selected components is low. We could consider that the all class-related information is contained within the first five components. The filtering operation is explained in Algorithm 4. Step 2 computes the variation mode and projection using least-squares estimate. We search $\widetilde{\mathbf{S}}_x$ supported by the K components $\widetilde{\mathbf{W}}$, given by Algorithm 4, which best fit with $\mathbf{S}_x$. We call this operation a filtering operation because there is no dimensional reduction.

Algorithm 4 Geodesic Filtering

Inputs: $\mathbf{P}_x$, $\mathbf{P}_{\mathbf{\Omega}}$, $\widetilde{\mathbf{W}} = [\widetilde{\mathbf{W}}_1 \dots \widetilde{\mathbf{W}}_K] \in \Re^{n(n+1)/2 \times K}$.
Output: $\widetilde{\mathbf{P}}_x$

1: $\mathbf{S}_x = \mathrm{Log}_{\mathbf{P}_{\mathbf{\Omega}}}(\mathbf{P}_x)$ {Apply Riemannian Log map}
2: $\widetilde{\mathbf{S}}_x = \widetilde{\mathbf{W}}\left(\widetilde{\mathbf{W}}^T\widetilde{\mathbf{W}}\right)^{-1}\widetilde{\mathbf{W}}^T vec\left(\mathbf{S}_x\right)$ {Filtering operation}
3: $\widetilde{\mathbf{P}}_x = \mathrm{Exp}_{\mathbf{P}_{\mathbf{\Omega}}}\left(unvec\left(\widetilde{\mathbf{S}}_x\right)\right)$ {Apply Riemannian Exp map}
4: **return** $\widetilde{\mathbf{P}}_x$

Finally, the core algorithm of this work is presented in Algorithm 5. First we compute FGDA filters on training dataset with Algorithm 3 , then we apply those filters on the dataset with Algorithm 4 and finally we use Algorithm 2 to obtain classes of filtered test data. It is also possible to apply a LDA classifier in tangent space without going back to original space. The LDA classifer uses only the first component and gives the same results as our complete method when taking one component.

Algorithm 5 Classification, filtered version

Inputs: Ω a set of m SPD matrices $\mathbf{P}_i \in P(n)$, $\omega_i \in [1:2]$ the class of $\mathbf{P}_i$, K number of kept components.
Input : $\mathbf{P}_x$ the SPD matrix to classify
Output: ω_x the class of $\mathbf{P}_x$

1: Compute FGDA filters (Algorithm 3)
2: Filter all SPD matrices $\mathbf{P}_i, \mathbf{P}_x$ (Algorithm 4)
3: Classify the filtered SPD matrices (Algorithm 2)
4: **return** ω_x

Figure 2 represents these different manipulations over a simulated dataset. 2×2 covariance matrices are generated according to two Wishart distributions, one for each classes. First, data are projected in tangent space using Log. map operator (Fig 2.D). Next, LDA components are computed and applied to the data (two components for Fig 2.E, only one for Fig 2.F). Finally, data are wrapped back to the original space with Exp. map operator (Fig 2.B and Fig 2.C).

4 Classification Results in BCI

4.1 Introduction

In order to evaluate the performances of the proposed methods, we compare them to an implementation of a reference method [8]. This latter method represents the classical signal processing chain in asynchronous BCI. It is composed of frequency filtering, spatial filtering (using CSP approach), Log Variance feature extraction

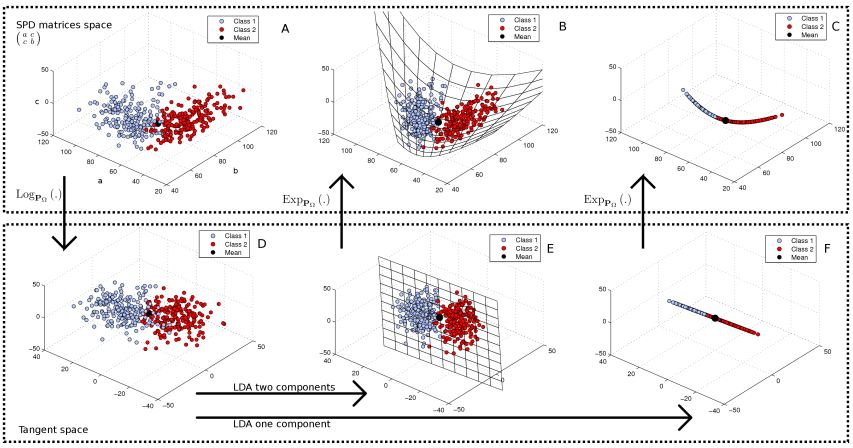

Fig. 2. Manipulations in tangent and original space of 2×2 covariance matrices

and finally Fisher's LDA classification. Datasets IVa of BCI competition III [10] are used for analysis. Only 9 electrodes are used (F_3, F_Z, F_4, C_3, C_Z, C_4, P_3, P_Z, P_4). This subset of electrodes can represent a typical case of an every-day use BCI. A $(10-30$ Hz) band-pass filter has been applied on the original signals for all subjects. This dataset is composed by 5 subjects who performed 280 trials of right-hand or right-foot motor imagery. It was provided by Fraunhofer FIRST, Intelligent Data Analysis Group. We use 10-fold cross-validation to evaluate properly the performance.

4.2 Classification Results

Results are given for both filtered (Algo 5) and unfiltered (Algo 2) version of the proposed classification algorithm based on Riemannian distance. We compare both methods with a reference method as described in Section 4.1. In the results, we kept only 4 FGDA filters for the filtered method and 6 CSP filters for the reference methods. Classification error rates are given in Table 1. The filtered version method (Algo 5) outperforms our implementation of the reference algorithm for almost all the subjects. The unfiltered version (Algo 2) is worst than the reference method but shows good results considering its simplicity. Because a high inter-subject variability, results are not really significant. However, we have benchmark our algorithms on several datasets and we can say that the Riemannian approach is effective and shows very good results in difficult cases (i.e. noisy dataset, small amount of data, multi-class).

5 Conclusion

We have presented an useful framework for working in space of SPD matrices, illustrating it with simple methods and giving results that show how promising

User	Reference	Algo 2	Algo 5
aa	26	28.9	**22.5**
al	3.2	3.9	**2.8**
av	34.2	39.6	**34.2**
aw	**6.4**	11	7.4
ay	7.4	12.1	**7.1**
Mean	15.5 ± 13.8	19.1 ± 14.7	$\mathbf{14.8} \pm 13.6$

Table 1. Classification Error rate. 10-fold Cross-validation

it is. This approach could be generalized to other signal processing methods in BCI or elsewhere. Methods can be developed to work natively in space of SPD matrices like our Algorithm 2 or in tangent space like Algorithm 3. The Euclidean Tangent Space allows us to use classical methods with the only limitation of the high number of dimension involved. In this work, FGDA is limited by the trend to over-fitting of the LDA algorithm, when the dimension is close to the number of trials. To avoid this effect a regularized LDA or a variable selection approach can be used. Furthermore, it is not always necessary to go back in SPD space, since the tangent space and original space are linked through Log. and Exp. map operators. Finally the main limitation of these methods is the computation time, Riemannian mean requires a large number of diagonalization of $n \times n$ matrices and and it becomes computationally expensive when n is large (> 50).

References

1. M.A Lebedev and M. A L Nicolelis. Brain-machine interfaces: past, present and future. *Trends in Neurosciences*, September 2006.
2. M. van Gerven, J. Farquhar, R. Schaefer, R. Vlek, J. Geuze, A. Nijholt, N.Ramsey, P. Haselager, L. Vuurpijl, S. Gielen, and P. Desain. The brain-computer interface cycle. *Journal of Neural Engineering*, August 2009.
3. J. R. Wolpaw, N.Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 2002.
4. P. T. Fletcher and S. Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*. 2004.
5. O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence*, 2008.
6. F. Barbaresco. Interactions between symmetric cone and information geometries: Bruhat-Tits and siegel spaces models for high resolution autoregressive doppler imagery. In *Emerging Trends in Visual Computing*. Springer-Verlag, 2009.
7. M. Moakher. A differential geometric approach to the geometric mean of symmetric Positive-Definite matrices. *SIAM J. Matrix Anal. Appl.*, 26, 2005.
8. B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller. Optimizing spatial filters for robust EEG Single-Trial analysis. *Signal Processing Magazine, IEEE*, 2008.
9. R. O. Duda, P. E. Hart, and D.G. Stork. *Pattern Classification*, chapter 3.8.2, pages 117–124. J. Wiley & Sons Inc, 2nd revised edition edition, November 2000.
10. BCI competition III, dataset IVa. http://ida.first.fhg.de/projects/bci/competition_iii.